



Lesson 33: Shiny

Ian Kloo

April 2019

Today

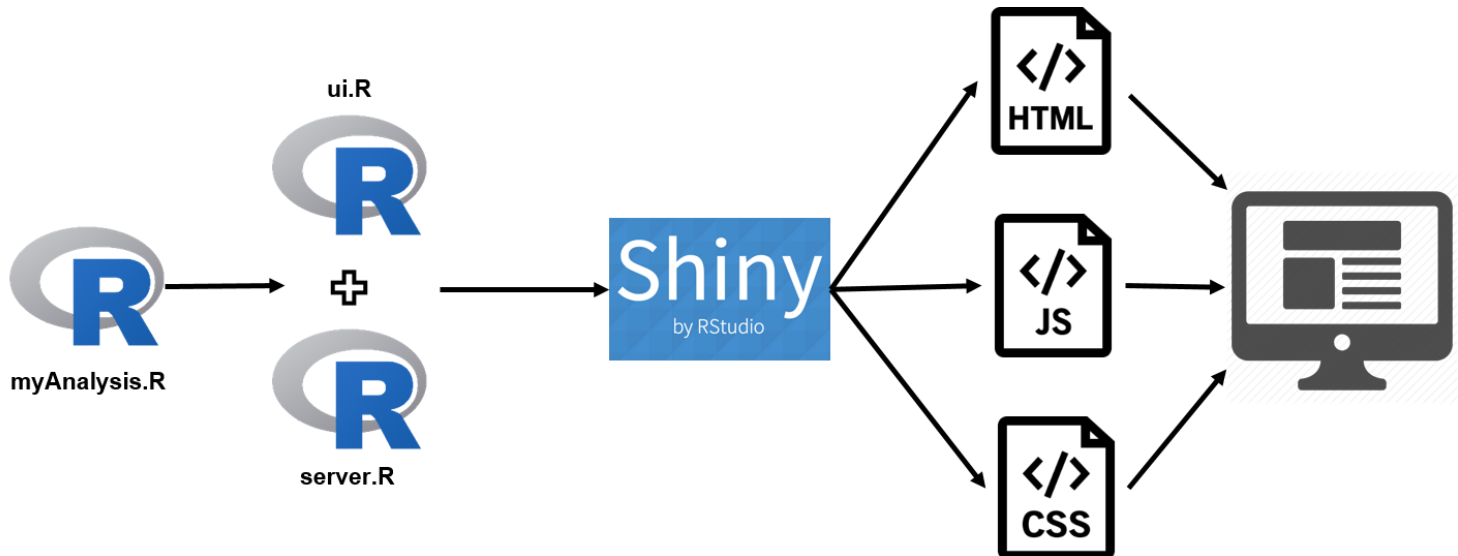
1. Application development in R with Shiny

What is Shiny?

- Shiny is an application (app) development framework for R.
- Using only R, you can write web applications!
- Think of Shiny as a way to give someone a GUI interface to your R code.
- Shiny is a RStudio product and they have a great website for reference:
<https://shiny.rstudio.com/>

How to make a Shiny app

- Write something useful in R
- Convert into Shiny syntax
- The Shiny package:
 - Converts your work to HTML, JS, and CSS
 - Starts a webserver
 - Deals with interactions between your R code and the app GUI



Shiny vs Rmarkdown

- Rmarkdown knits output to HTML
- Shiny also creates HTML, but keeps a line of communication to R



VS



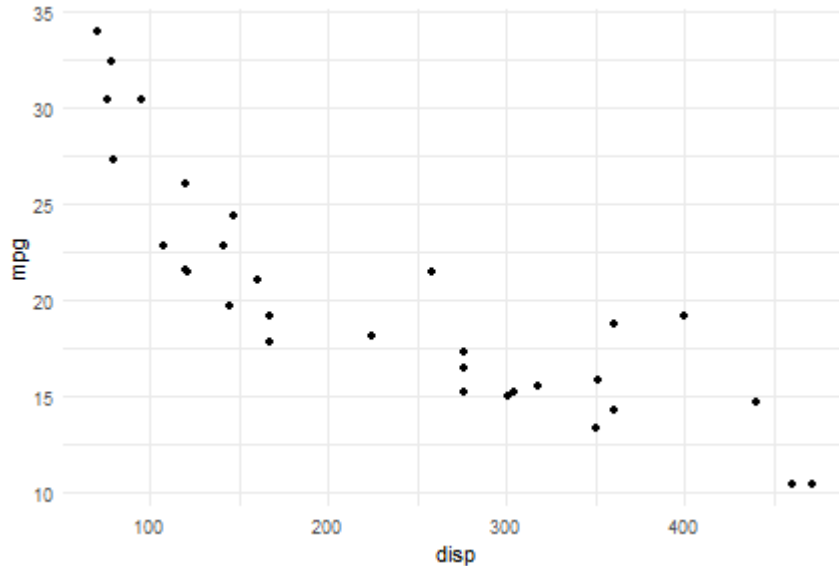
Other Shiny thoughts

- Shiny allows people to run R code by just clicking on things
- This is a great way to deliver more complicated analysis, or things with varying inputs
- You need a running R session or a Shiny server to run your Shiny app (you can't just email it to someone)
- We will discuss deploying apps next class (you can do it on MatrixDS)

Let's make an app!

- Let's start with a simple ggplot:

```
library(ggplot2)
ggplot(data = mtcars, aes(x = disp, y = mpg)) + geom_point() +
  theme_minimal()
```



Make this into an app!

- Make a new folder for your app...call it whatever you want
- Copy/paste this into a file called ui.R

```
shinyUI(fluidPage(  
  plotOutput(outputId = "myplot", height = "300px", width = "700px")  
))
```

- And this into a file called server.R

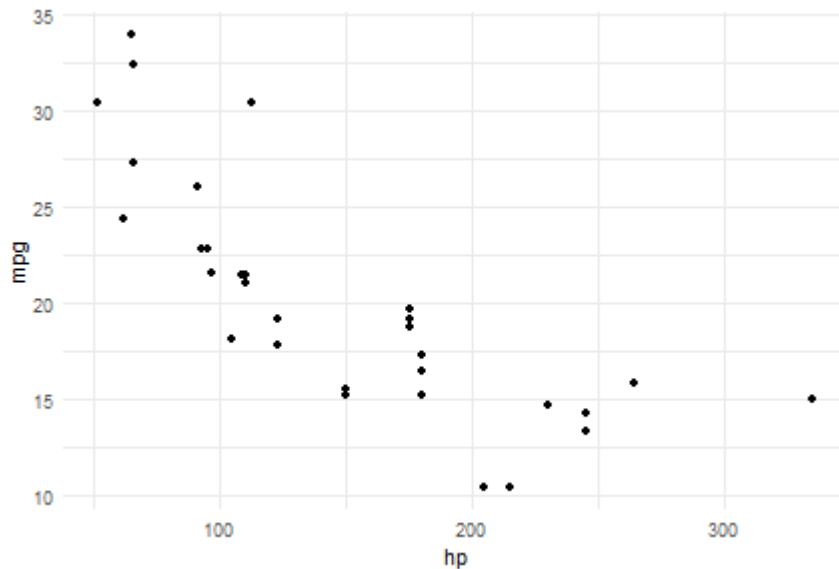
```
library(ggplot2)  
shinyServer(function(input, output) {  
  output$myplot <- renderPlot({  
    ggplot(data = mtcars, aes(x = disp, y = mpg)) +  
      geom_point()  
  })  
})
```

- Click "Run App"

Change something

- We can change the variable we show on the x axis easily...

```
ggplot(data = mtcars, aes(x = hp, y = mpg)) + geom_point() +  
  theme_minimal()
```



Make a dropdown in the app to pick the x axis variable

- `selectInput` creates a dropdown

ui.R

```
shinyUI(fluidPage(  
  plotOutput(outputId = "myplot", height = "300px"),  
  selectInput(inputId = "xAxis", label = 'Select x axis variable',  
              choices = colnames(mtcars))  
))
```

Link the input to the server function

- `input$xAxis` holds the value selected using our dropdown

server.R

```
library(ggplot2)
shinyServer(function(input, output) {
  output$myplot <- renderPlot({
    ggplot(data = mtcars, aes_string(x = input$xAxis, y = 'mpg')) + geom_point()
  })
})
```

Your turn!

- Use the same technique to create a dropdown for the y-axis as well

Add a slider to change point size

ui.R

```
shinyUI(fluidPage(  
  plotOutput(outputId = "myplot", height = "300px"),  
  selectInput(inputId = "xAxis", label = 'Select x axis variable',  
             choices = colnames(mtcars)),  
  selectInput(inputId = "yAxis", label = 'Select y axis variable',  
             choices = colnames(mtcars)),  
  sliderInput(inputId = 'pointSize', label = 'Slide to change the point size',  
             min = 1, max = 10, value = 2)  
))
```



Link the slider to the server function

server.R

```
library(ggplot2)
shinyServer(function(input, output) {
  output$myplot <- renderPlot({
    ggplot(data = mtcars, aes_string(x = input$xAxis, y = input$yAxis)) +
      geom_point(size = input$pointSize)
  })
})
```

Your turn!

- Figure out how to change the color of the points in a regular ggplot (without worrying about shiny)
- Figure out how to add radio buttons to your ui.R file. (Hint: <https://shiny.rstudio.com/gallery/>)
- Link the input value of those radio buttons to the server.R file